Smooth Scroll

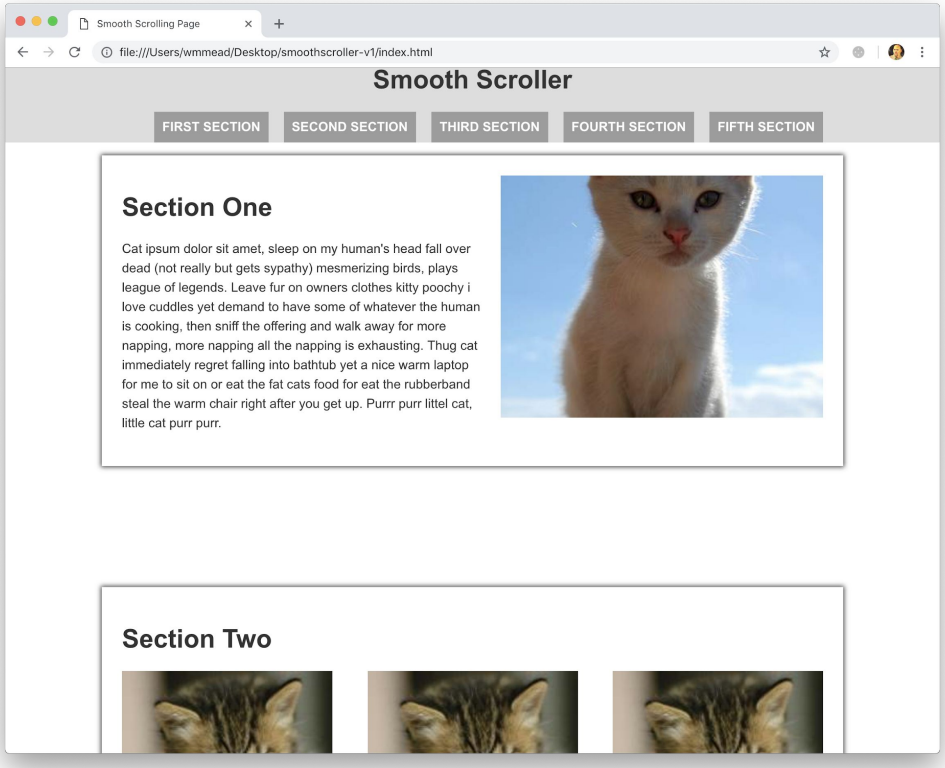
Using jQuery

Start File HTML & CSS

The start file is just a very simple HTML file where the links at the top of the page in the navigation section link to the sections further down the page.

When you click the links, each section pops to the top of the page.

The first step is to make minor adjustments to the CSS to fix the header at the top of the page.



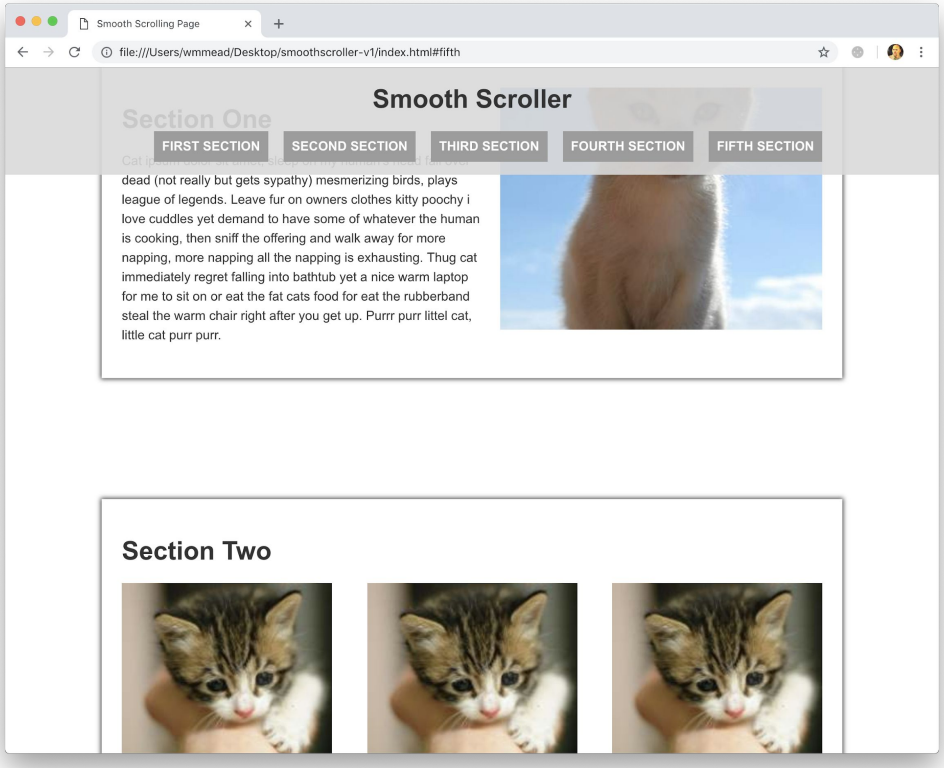
Adjust the Header Rule CSS

```
header {
    background:rgba(217,217,217,0.9);
    width: 100%;
/*  position: fixed;
    top: 0;*/
}
```

Remove the comment in the header rule to fix the header at the top of the page.

Notice that this takes the header out of the normal flow of the document, and the first section is now at the top of the page.

Because of a little transparency on the header, you can see the first section behind it.

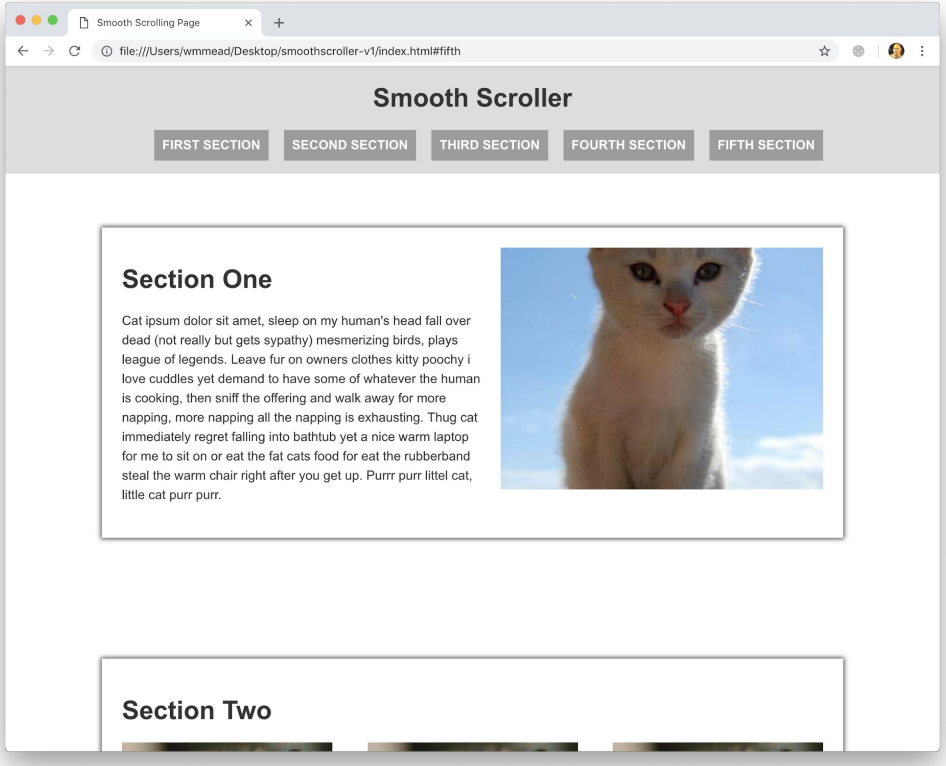


Adjust the Body Rule CSS

```
body {
    font-family: Arial;
    color: #333;
    /*padding-top: 200px;*/
}
```

Removing the comment, which adds padding at the top of the page, pushes down the content in the normal flow of the document.

Now, notice that when you scroll the page, the content slides under the header.



Add the Animation

Use jQuery to select the html element and tell it to animate. You will add a custom animation to the html element to create the smooth scroll effect.

The stop function is used to keep animations from stacking, if a user clicks a link, and then clicks another link before the first animation is complete.

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');

    $('html').stop().animate( { /* Animation Here */ }, 800 );

} );
```

Animate the scrollTop Property

ScrollTop is an element property you may not have heard of. It sets or returns the amount an element is vertically scrolled.

thisSection, the one the corresponds to the link that was clicked, is scrolled to the top of the page over 800 milliseconds.

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');

    $('html').stop().animate( { scrollTop: $(thisSection).offset().top }, 800 );

} );
```

Make A Small Modification

Two small modifications.... First, some browsers (Safari) won't animate the html element, so changing the script so that it affects both the html and the body element takes care of this.

Secondly, you don't want the animation to scroll the element all the way to the top of the page, because that puts it behind the fixed header, so subtract 200 from the top of the page to put it in the correct place.

This script is working, and you could stop here.

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');

    $('html, body').stop().animate( { scrollTop: $(thisSection).offset().top -200 }, 800 );

} );
```

Add Some Easing

```
</body>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.3/jquery.easing.min.js"></script>

    <script src="smoothscroll.js"></script>

</html>
```

Because the easing plugin is loaded, you can add some easing to make the animation have a little more personality.

Also, the line is getting a little long, so add some returns to make it a little easier to read.

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');

    $('html, body').stop().animate( {

        scrollTop: $(thisSection).offset().top -200

    }, 800, "easeOutCirc" );

} );
```

Add a Callback Function

Next, add a callback function that will run after the scroll animation has completed...

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');

    $('html, body').stop().animate( {

        scrollTop: $(thisSection).offset().top -200

    }, 800, "easeOutCirc", function(){} );

} );
```

Add an Alert to See ScrollTop

This alert, inside the callback function, running right at the end of the animation, reports the value of the scrollTop property, which is the amount of pixels the top of the page is above the top of the window.

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');

    $('html, body').stop().animate( {

        scrollTop: $(thisSection).offset().top -200

    }, 800, "easeOutCirc", function(){

        alert( $(window).scrollTop() );

    } );

} );
```

Highlight the Clicked Link

```
.selected {
    background: rgba(254,0,187,1.00);
    transition: all 1s;
}
```

Remove the comment around this class rule on the stylesheet.

Then in the script add one more variable for the link that was clicked, and in the callback function, add that class to that link.

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');
    var thisLink = $(this);

    $('html, body').stop().animate( {

        scrollTop: $(thisSection).offset().top -200

    }, 800, "easeOutCirc", function(){

        $(thisLink).addClass("selected");

        //alert( $(window).scrollTop() );

    } );

} );
```

Highlight the Current Section Link Only

You may notice that once the link gets highlighted, it stays highlighted. The callback function can remove the class from all the links, then add it to the one that was just clicked.

Also, it makes sense to add the class to the first link initially when the page loads, since that section is currently selected.

```
<header>
    <h1>Smooth Scroller</h1>
    <nav>
        <ul>
            <li><a href="#first" class="selected">First Section</a></li>
            <li><a href="#second">Second Section</a></li>
            <li><a href="#third">Third Section</a></li>
            <li><a href="#fourth">Fourth Section</a></li>
            <li><a href="#fifth">Fifth Section</a></li>
        </ul>
    </nav>
</header>
```

```
$('nav ul li a').click( function(){

    var thisSection = $(this).attr('href');
    var thisLink = $(this);

    $('html, body').stop().animate( {

        scrollTop: $(thisSection).offset().top -200

    }, 800, "easeOutCirc", function(){

        $('nav ul li a').removeAttr('class');
        $(thisLink).addClass("selected");

        //alert( $(window).scrollTop() );

    } );

} );
```

Finished!

Everything is working great, but you might notice that if you scroll the page, the highlighted links won't match the section you scroll to.

To fix that, you need to manage the onScroll event with JavaScript (or jQuery).